CLAIMS

What is claimed:

1. A method for monitoring memory usage by a software agent executing in a computer system, said method comprising:

starting a resource tracking function for monitoring information associated with said software agent in machine-executable code;

creating a computer-readable data structure for storing information about said agent;

identifying a process operating on said system and to which said agent is operatively associated;

determining memory usage data for said agent; and

storing said memory usage data in said data structure.

2. The method of claim 1, wherein said computer-readable data structure is a hash table.

3. The method of claim 2 wherein said process is identified as being a non-hypertext transport protocol (non-HTTP) process.

4. The method of claim 3 wherein said determining step further includes:

creating a thread list for monitoring threads associated with said agent;

identifying at least one thread associated with said agent to produce at least one identified thread;

adding said at least one identified thread to said thread list;

determining a memory allocation associated with said at least one identified thread to produce at least one determined allocation for said at least one identified thread; and

computing a peak memory usage for said agent using said at least one determined allocation;

thereby monitoring memory usage by a software agent executing in a computer system having a non-HTTP process operating thereon.

5. The method of claim 4 further comprising:

comparing said peak usage for said agent to a plurality of peak usages associated with a like plurality of other agents executing in said system.

6. The method of claim 5 further comprising:

computing statistics on said agent and said plurality of other agents; and

ranking said agent against said plurality of other agents based on said peak usage to produce a ranked list.

7. The method of claim 6 further comprising:

displaying said ranked list to a user.

8. The method of claim 2, wherein said process is identified as being an HTTP process.

9. The method of claim 8, wherein said determining step further includes:

identifying HTTP threads operating in said system to produce identified threads, each of said identified threads further having one of a plurality of agent types associated therewith, at least one of said plurality of agent types including said agent, said agent capable of having agent threads associated therewith;

generating an agent thread list for facilitating identification of said plurality of agent types by storing information associated therewith;

identifying which of said plurality of agent types is operating on each of said HTTP threads;

associating those of said agent threads said agent is running on together to produce a related agent set;

determining memory usage for each thread in said related agent set; and

combining said memory usage for each thread in said related agent set to produce a total memory consumption for said agent;

thereby monitoring memory usage by a software agent executing in a computer system having an HTTP process operating thereon.

10. The method of claim 9 further comprising;

comparing said total memory consumption for said agent to a like plurality of total memory consumptions associated with others of said plurality of agents types.

11. The method of claim 10 further comprising:

computing statistics on said total memory consumption for said agent and each one of said plurality of total memory consumptions; and

ranking said agent against said others of said plurality of agent types using said total memory consumption and said plurality of total memory consumptions, respectively, to produce a ranked list.

12. The method of claim 11 further comprising:

displaying said ranked list to a user.

13. The method of claim 1, further comprising:

determining if said agent is running before determining said memory usage.

14. The method of claim 13, further comprising:

determining if said agent is expired; and

processing said information if said agent is expired.

15. The method of claim 1, wherein said data structure further includes information about a plurality of other software agents.

16. The method of claim 15, further comprising:

establishing a threshold for maximum memory usage.

17. The method of claim 16, further comprising:

terminating said software agent and those of said plurality of other software agents exceeding said threshold.

18. A computer program product having machine-readable instructions disposed thereon for instructing a processor to perform a method for identifying memory usage information associated with a software agent operating in a computer system, said computer program product comprising:

instructions for initiating resource tracking function-executable instructions for monitoring information associated with said software agent;

instructions for generating a computer-readable data structure residing in computer-accessible memory for storing said memory usage information associated with said agent;

instructions for determining said memory usage information; and

instructions for storing said memory usage data in said computer-readable data structure.

19. The computer program product of claim 18, further comprising:

instructions for storing memory usage information about a plurality of other software agents;

instructions for processing said memory usage information associated with said agent and said memory usage information about said plurality of other software agents; and

instructions for generating a rank order list including said information about said agent and said information about said plurality of other software agents.

20. An apparatus for monitoring operation of a software agent, said apparatus comprising:

a processor executing machine-readable instructions for starting a resource tracking function for monitoring and processing information associated with a software agent operating in connection with said apparatus; and

a memory communicatively associated with said processor for storing information about memory usage of said agent in a data structure;

thereby monitoring said operation of said software agent.